

Hackathon Management System

A comprehensive Java-based management system for organizing and scoring hackathon teams, implementing three-tier architecture and MVC design pattern.

Table of Contents

- [Features](#)
- [Architecture](#)
- [Project Structure](#)
- [Requirements](#)
- [Installation & Running](#)
- [Usage](#)
- [Class Descriptions](#)
- [Assignment Compliance](#)

Features

Core Functionality

- **Team Management:** Add, edit, delete, and view teams
- **Multiple Categories:** Support for Cybersecurity, AI, Web Development, and Mobile Development
- **Smart Scoring:** Different scoring algorithms per category
- **File I/O:** CSV import/export functionality
- **Comprehensive Reporting:** Statistics and detailed reports
- **Search Functionality:** Quick team lookup by number

User Interfaces

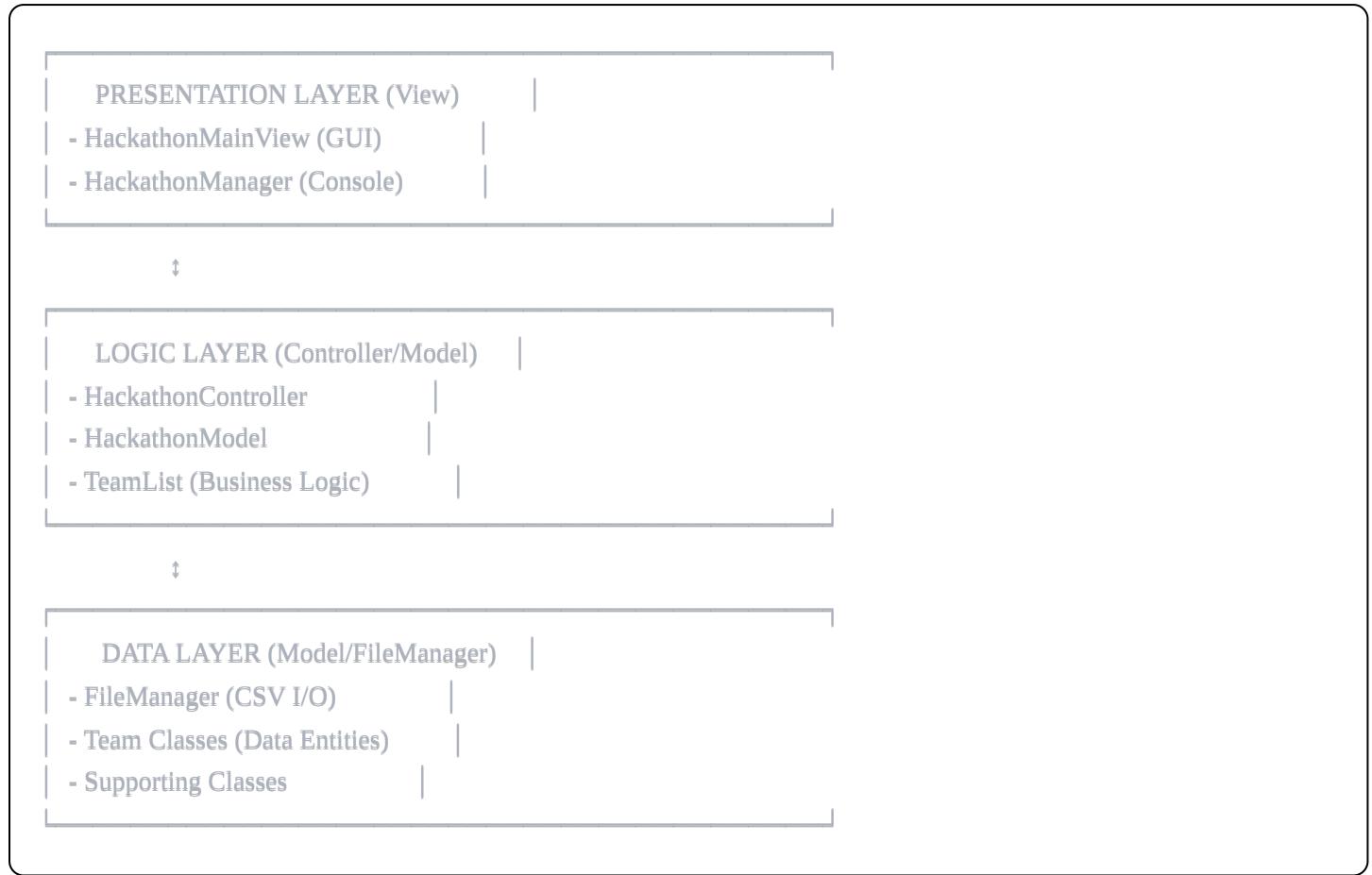
- **GUI Interface:** Full-featured Swing-based interface with MVC pattern
- **Console Interface:** Text-based interface for testing

Advanced Features

- **Inheritance:** Abstract Team base class with category-specific subclasses
- **ArrayList Management:** Efficient team data handling
- **Data Validation:** Input validation for scores and team information
- **Report Generation:** Automated report creation and file export

Architecture

Three-Tier Architecture



MVC Design Pattern

Model:

- `HackathonModel`: Manages application data and business logic
- `TeamList`: Handles team collection operations
- `Team` and subclasses: Data entities

View:

- `HackathonMainView`: GUI interface with multiple tabs
- Panels for viewing, searching, editing, and statistics

Controller:

- `HackathonController`: Mediates between Model and View
- Handles user actions and updates

Project Structure

HackathonManagementSystem/

```

├── Team.java           # Abstract base class
├── CybersecurityTeam.java # Subclass
├── AITeam.java         # Subclass
├── WebDevTeam.java     # Subclass
└── MobileAppTeam.java  # Subclass

|
├── TeamList.java        # ArrayList manager
└── FileManager.java     # File I/O handler

|
├── HackathonModel.java   # MVC Model
├── HackathonMainView.java # MVC View (GUI)
├── HackathonController.java # MVC Controller
├── HackathonManagementSystem.java # Main GUI application
└── HackathonManager.java  # Console application

|
├── Category.java        # Supporting classes
├── Judge.java
├── Score.java
├── TeamMember.java
└── User.java

|
├── HackathonTeams.csv    # Sample data file (auto-generated)
└── README.md             # This file

```

Requirements

- Java Development Kit (JDK) 8 or higher
- No external libraries required (uses standard Java libraries)

Installation & Running

Compile All Files

```

bash
javac *.java

```

Run GUI Application

```

bash
java HackathonManagementSystem

```

Run Console Application

```

bash

```

Usage

GUI Application

The GUI provides four main tabs:

1. View Teams Tab

- Display all teams in a table
- Filter by category
- Sort by team number, name, or score
- Refresh button to update display

2. Search Team Tab

- Enter team number to search
- View full team details
- Quick lookup functionality

3. Edit Team Tab

- Load team by number
- Modify team information
- Update scores with instant overall score calculation
- Delete teams
- Save changes

4. Statistics Tab

- View summary statistics
- Category breakdown
- Score frequency analysis
- Generate full report to text file

Console Application

Menu options:

1. Display All Teams
2. Search Team by Number
3. Display Highest Scoring Team

4. Display Statistics
5. Generate Full Report
6. Add New Team
7. Edit Team
8. Delete Team
9. Save Teams to File
10. Exit

CSV File Format

```
TeamNumber,TeamName,University,Category,Score1,Score2,Score3,Score4,Score5  
101,Cyber Titans,MIT,Cybersecurity,3,4,3,4,4  
102,AI Innovators,Stanford,Artificial Intelligence,5,4,5,4,3
```

Class Descriptions

Core Classes

Team (Abstract)

- Base class for all team types
- Contains common attributes and methods
- Abstract `getOverallScore()` method

Team Subclasses Each subclass implements its own scoring algorithm:

- **CybersecurityTeam:** Average of top 4 scores
- **AITeam:** Weighted average (weights: 1, 1.5, 2, 1.5, 1)
- **WebDevTeam:** Average excluding highest and lowest
- **MobileAppTeam:** Simple average

TeamList

- Manages ArrayList of teams
- Provides search, filter, and statistical methods
- Generates reports

FileManager

- Handles CSV reading and writing

- Creates sample data files
- Manages report generation

MVC Components

HackathonModel

- Central data repository
- User authentication
- Team, judge, and category management

HackathonMainView

- Swing-based GUI with tabbed interface
- Table displays and form inputs
- Event handling

HackathonController

- Coordinates Model and View
- Implements business logic
- Handles user actions

Supporting Classes

Judge: Manages judge information and category assignments **Score:** Detailed scoring with multiple criteria

Category: Hackathon category definitions **TeamMember:** Team member information **User:** Authentication and authorization

Assignment Compliance

Stage 4: Basic Implementation

-  Team class with constructor, getters, setters
-  getOverallScore() method
-  getFullDetails() method
-  getShortDetails() method with format "TID X (XX) has an overall score of X.XX"
-  Array of integer scores (0-5)
-  Main method for testing

Stage 5: Enhanced Data Handling

-  TeamList class with ArrayList

- 10-15 teams with varied data
- CSV file reading (HackathonTeams.csv)
- Report generation to text file
- Full details table
- Highest scoring team display
- Four summary statistics (total, average, max, min)
- Frequency report for individual scores
- Team search by ID with validation
- Input file error checking

Stage 6: Inheritance & GUI

- Abstract Team superclass
- Multiple Team subclasses with different implementations
- Refactored TeamList to work with all Team types
- Updated class diagram reflecting inheritance
- GUI with multiple panels/windows
- View and alter team scores
- Sort teams in different ways
- Filter teams by category/attributes
- View full/short details by team number
- Edit team details
- Remove teams
- MVC design pattern implementation
- Close button generates report and exits
- No drag-and-drop IDE used

Additional Requirements

- Three-tier architecture
- Judge, Score, Category, TeamMember classes
- User authentication system
- Admin, Judge, Participant role support

-  Comprehensive documentation

Key Features by User Role

Admin

- Manage all teams
- View comprehensive statistics
- Generate reports
- Full CRUD operations

Judge

- View assigned categories
- Enter and update scores
- Calculate overall scores

Participant

- Register teams
- Update team details
- View scores

Public

- View results
- Search teams
- View leaderboards

Notes

- The system automatically creates a sample CSV file on first run
- All data modifications are saved when exiting the GUI application
- Scores must be between 0 and 5 (validation enforced)
- Each team category uses a different scoring algorithm
- Reports are saved to HackathonReport.txt

Support

For questions or issues, please refer to:

- The inline code comments

- The class documentation
 - The usage examples in this README
-

Project Status:  Complete - All assignment requirements met **Version:** 1.0 **Last Updated:** December 2024